

# Symcloud

Filehostingplattformen sind in der heutigen Zeit allgegenwärtig. Ohne einen Zugriff zu einem der allgemein verfügbaren Dienste, ist heutzutage eine Zusammenarbeit in einer Gruppe von Menschen zum Beispiel an einer Universität kaum möglich.

In den letzten Jahren wurde allerdings, nicht zuletzt durch den Wistle Blower Edward Snowden, bekannt dass genau diese Daten von verschiedensten Quellen abgesaugt und verarbeitet werden.

Diese Angst vor Kontrollverlust bewegt viele Menschen dazu Ihre Daten am liebsten bei sich zu behalten und mit Quelloffenen Lösungen die Daten auf Ihrem eigenen Server zu speichern.

TODO Vorstellung? und Agenda

## Motivation

Meine Masterarbeit beschäftigt sich mit der Konzeption einer Speicherlösung, die Ideen aus verschiedenen Applikationen und Technologien kombiniert und es damit den Anwendern ermöglichen soll Ihre Daten lokal und sicher zu speichern. Diese aber bei Bedarf gezielt an andere Benutzer weiterzugeben, selbst wenn diese nicht auf dem selben Server registriert sind.

## Einführung

Die Idee entstand vor ein paar Jahren als ich auf ein Open-Source Projekt namens ownCloud gestoßen bin und reifte dann in den letzten Monaten zu meiner Masterarbeit heran.

ownCloud ist eine freie Software für das Speichern von Daten (Filehosting) auf einem eigenen Server. Bei Einsatz eines entsprechenden Clients wird dieser automatisch mit einem lokalen Verzeichnis synchronisiert. Über eine Plattform lassen sich die Daten auch im Browser bearbeiten und verwalten.

Zuerst ging es bei dem Projekt lediglich um eine Plattform die es ähnlich wie oenCloud ermöglichen soll, Dateien zu verwalten und mit anderen zu teilen. Bei der Recherche jedoch erkannte ich, dass die Unabhängigkeit von einem Provider sehr wichtig für die Zielgruppe des Projektes ist. Jeder der Anwender will sein eigener Provider sein. Jedoch ist es dann bei klassischen Systemen wie ownCloud oder anderen nur schwer möglich Dateien über die Grenzen des eigenen Servers zu teilen.

Daher wurde der Fokus zu Beginn der Projektzeit sehr schnell auf genau diesen Anwendungsfall gelegt.

Auf der suche nach Referenzprojekten bin ich schnell auf zwei sehr starke Inspirationsquellen gestoßen. Beide Projekte sind in ihrer Zeit gesehen Pilotprojekte und Vorreiter ihrer Technologien.

## Projekt Xanadu

Das Projekt Xanadu wurde durch Theodor Holm Nelson - kurz Ted - in den 1960er Jahren initiiert und ist bis heute nicht finalisiert worden. Jedoch prägte Ted Nelson mit der Gründung des Projektes und durch die Veröffentlichung des wissenschaftlichen Artikels "The Hypertext" im Jahre 1965 den Begriff des Hypertext und inspirierte damit unter anderem Tim Berners-Lee zu der Entwicklung des WorldWideWeb mehr als zwei Jahrzehnte später.

Die Kernaussage des Projektes beinhalten 12 Thesen, die zum Teil im heutigen Web Anwendung finden. Einige dieser Thesen, die nicht oder nur zum Teil umgesetzt wurden, sind für symCloud sehr Interessant und werden in der Arbeit genauer betrachtet.

Die Kernaussage dieser 6 ausgewählten Thesen sind:

- Jeder Server kann unabhängig und in einem Verbund von gleichwertigen Servern arbeiten.
- Jeder Benutzer ist sicher und eindeutig identifizierbar.
- Jeder Benutzer kann Dokumente durchsuchen, anlegen, hoch und herunterladen.
- Jedes Dokument besitzt eine Zugriffskontrollliste über die entschieden werden kann welcher Benutzer zugriff besitzt.
- Jedes Dokument kann verwendet werden ohne genau zu wissen wo es physikalisch gespeichert wurde.
- Jedes Dokument ist zur Sicherheit redundant gespeichert.

## **TODO Aussage**

## **Diaspora**

Diaspora ist ein Dezentrales Soziales Netzwerk. Es ist vergleichbar mit Facebook welches jedoch zentralisiert aufgebaut ist. Was bedeutet, dass jeder Benutzer der mit anderen kommunizieren will auch bei Facebook registriert sein muss.

Diaspora geht hier einen anderen Weg. Über ein spezielles Protokoll können z.B. Nachrichten auch mit Benutzern ausgetauscht werden, die auf einem anderen Diaspora Server registriert sind.

## **Ziele**

Als Ziel der Arbeit wurde ein funktionsfähiges und erweiterbares Konzept festgelegt, dass die Vorteile der eben beschriebenen Projekte vereint. Aus diesem Konzept sollte ein Prototyp entstehen, der die wichtigsten Komponenten des Konzeptes beinhaltet.

## **Anforderungen**

Aus den drei Inspirationsquellen wurden die Anforderungen an ein solches System formuliert in drei Teilgebiete unterteilt.

1. Datensicherheit
2. Filehosting und Filesharing Funktionalitäten
3. Architektur

Zusätzlich wurde darauf geachtet, dass die Entwicklung auf dem aktuellen Stand der Technik und mit einem Fokus auf Wartbarkeit und Erweiterbarkeit umgesetzt wird.

## **Konzept**

Um ein ausgewogenes Konzept zu erstellen, wurden einige Technologien aus den Bereichen

- Verteilte Datenmodelle - GIT
- Verteilte Daten - Diaspora
- Verteilte Dateisysteme - NFS und XtremFS
- Objekt - Speicherdienste
- Datenbankgestützte Dateiverwaltung - GridFS

evaluiert.

Die jeweiligen Vorteile dieser Systeme wurden in symCloud kombiniert. Um nur ein paar Beispiele zu nennen:

- Das Datenmodell von GIT erweitert um die Anforderungen vollständig zu erfüllen.
- Die lose gekoppelte Server-Architektur von Diaspora findet auch Anwendung in der Architektur von symCloud.
- Mithilfe einer einfachen Variante des Primärbasierten-Backup-Protokolls von XtremFS werden die Objekte in einem Netzwerk von Servern verteilt. Diese Verteilung wird unter anderem von den Benutzerrechten gesteuert. Was bedeutet, dass diejenigen Server eine Kopie der Daten erhalten, auf denen mindestens ein Benutzer registriert ist, der Zugriff auf die Daten besitzt.

TODO Ergebniss aussage?

## Implementierung

Bei der Implementierung wurde auf bewährte Technologien aus dem Umfeld von PHP und dem Framework Symfony2 vertraut. Auf diese Technologien möchte ich hier nicht genauer eingehen. Dies würde den Rahmen dieser Präsentation sprengen.

Entstanden ist während der Projektzeit ein funktionstüchtiger Prototyp, der die Funktionalen Anforderungen erfüllt.

Es wurde eine Bibliothek implementiert, die eine einfache verteilte Datenbank enthält, die die Daten lokal in einem Ordner ablegt und über das Verteilungsprotokoll die Daten an andere bekannte Server repliziert. Diese Daten können dann von dem anderen Server verwendet werden.

Diese Bibliothek wurde in eine bestehende Content Management Plattform integriert. Über eine einfache Benutzerschnittstelle, können die Benutzer die Daten bearbeiten bzw. erstellen.

Als zusätzliche Komponente wurde ein Synchronisierungstool umgesetzt, mit dem es möglich ist die Daten von einem lokalen Ordner mit einem Server zu synchronisieren. Dieses Tool erkennt automatisch welche Dateien aktualisiert, erstellt oder gelöscht werden müssen. Diese Funktionalitäten funktionieren bidirektional. Das bedeutet, dass das Tool beidseitig Änderungen erkennen kann.

## Probleme

Während der Entwicklung des Konzeptes stellte sich die Kombination aus Immutable (also unveränderbare) und nicht Immutable Objekten, die im Datenmodell von GIT vorhanden sind, als Problem heraus. Die Datenbank wurde aus diesem Grund während der Entwicklungszeit immer aufwändiger und komplexer.

Für eine Weiterentwicklung des Konzeptes wäre zu Überlegen diese Objekte gesondert zu speichern. Dadurch könnte vieles an Komplexität aus der Datenbank entfernt werden.

Aktuell stellt die Performance des Prototypen und im speziellen des Verteilungsprotokolls die größten Herausforderungen dar. Unter anderem könnte die Implementierung eines "Diff" Algorithmus einen Performancegewinn durch geringeren Datentransfers erreichen. Einen Lösungsansatz wird ebenfalls in der Arbeit kurz beschrieben.

## Fazit

Das in der Arbeit entwickelte Konzept stellt eine gute Grundlage für weitere Entwicklungen dar. Es umfasst alle Facetten von Filesharing und -hosting. Durch die Flexibilität ist es möglich auch Große Dateien effizient zu speichern, obwohl sich auch hier Verbesserungsmöglichkeiten während der Projektzeit aufgetan haben.

Interessant wäre es auch aus diesem Konzept eine eigenständige Datenbank mit einer wohldefinierten Schnittstelle zu schaffen. Diese Datenbank könnte sich dann um alle Facetten des Filehostings kümmern.

**TODO umformulieren (Inhalt ist aber gut)**